Title: METHOD AND APPARATUS FOR HIGH-PERFORMANCE RENDERING AND HIT-TESTING OF A WINDOW TREE
Inventor: Jeffrey E. Stall
Docket No.: MSFT116683

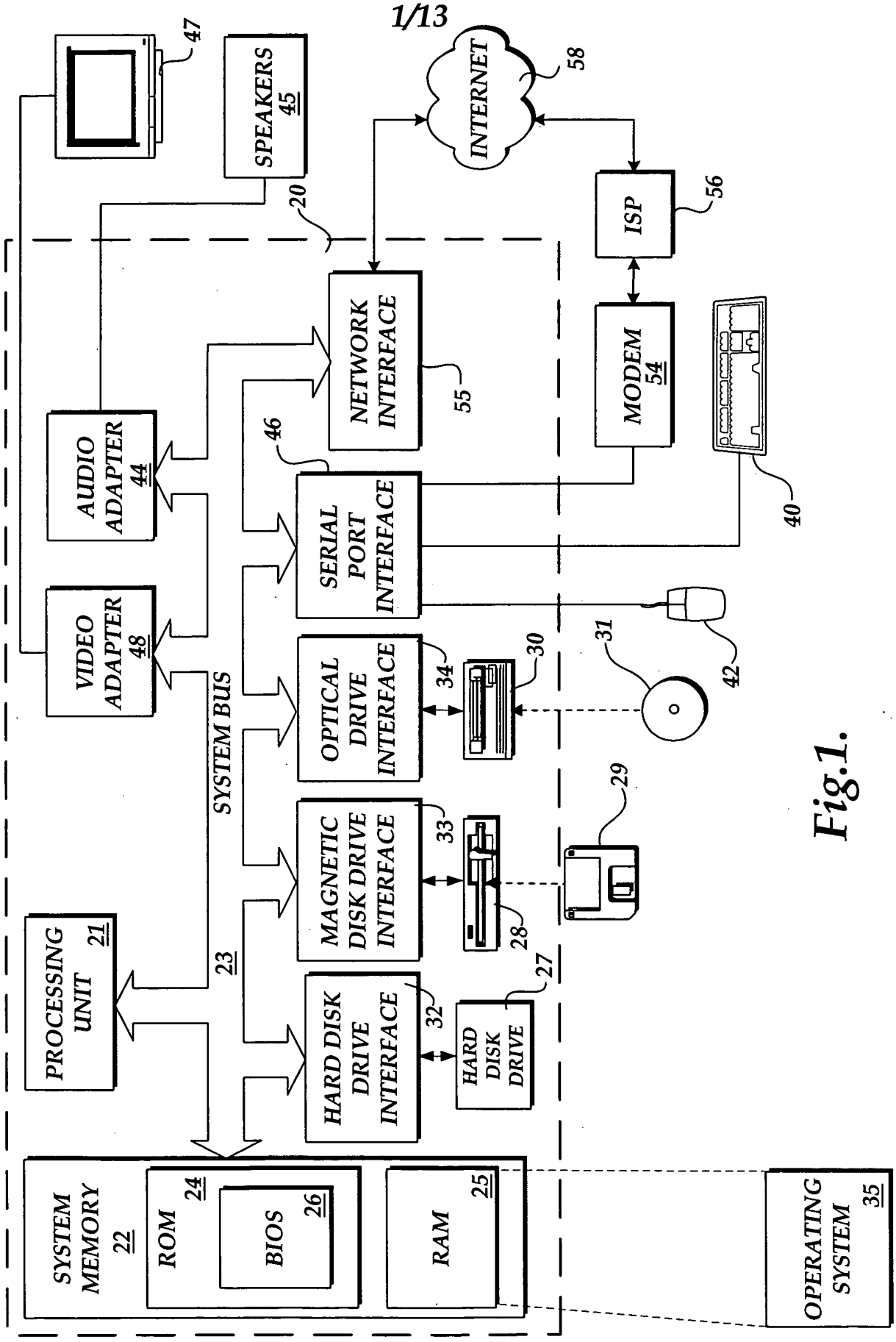

Fig.1.

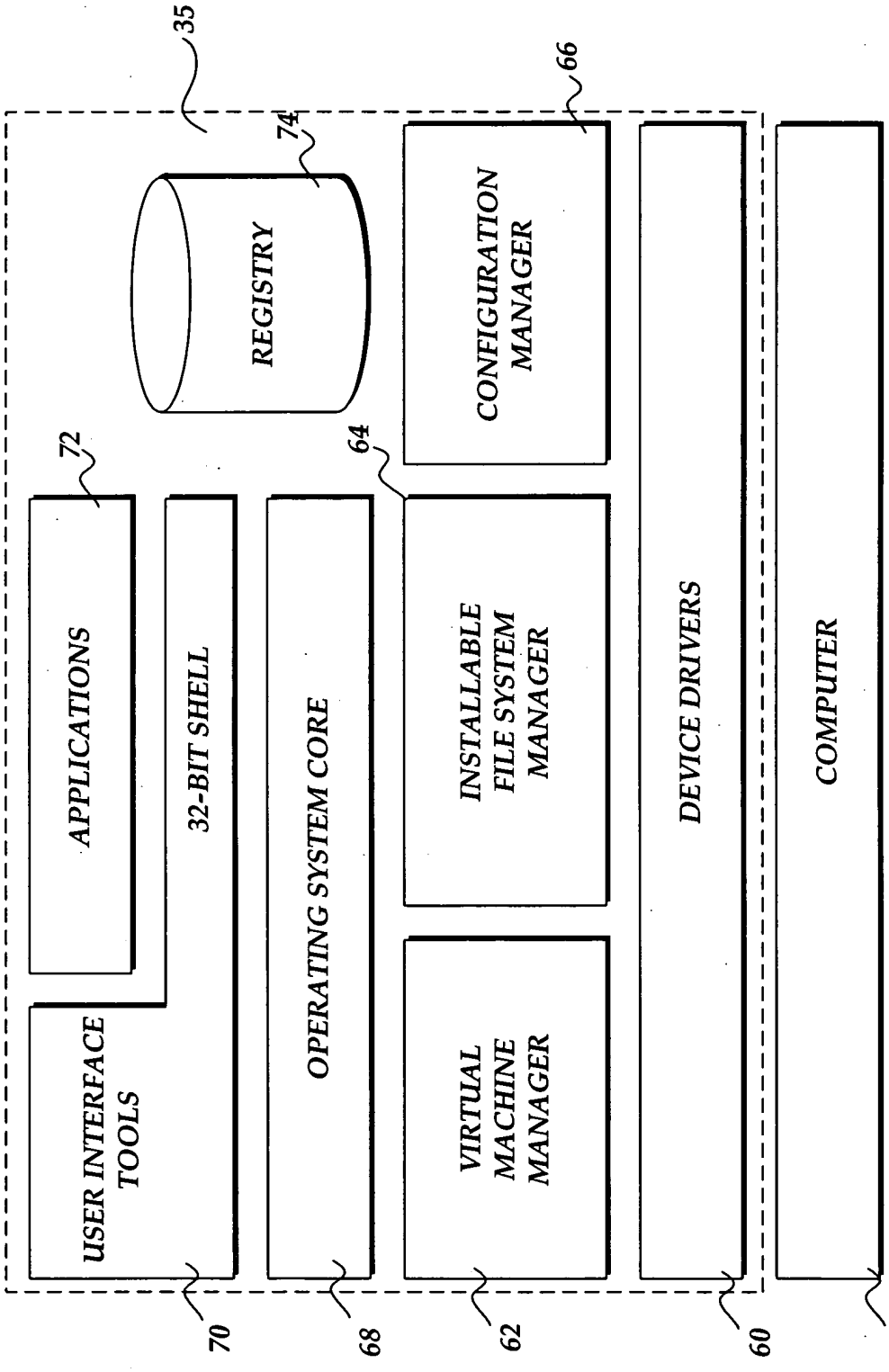
35
66
74
REGISTRY
CONFIGURATION MANAGER
72
64
APPLICATIONS
32-BIT SHELL
OPERATING SYSTEM CORE
INSTALLABLE FILE SYSTEM MANAGER
DEVICE DRIVERS
COMPUTER
USER INTERFACE TOOLS
VIRTUAL MACHINE MANAGER
70
68
62
60
20

Fig.2

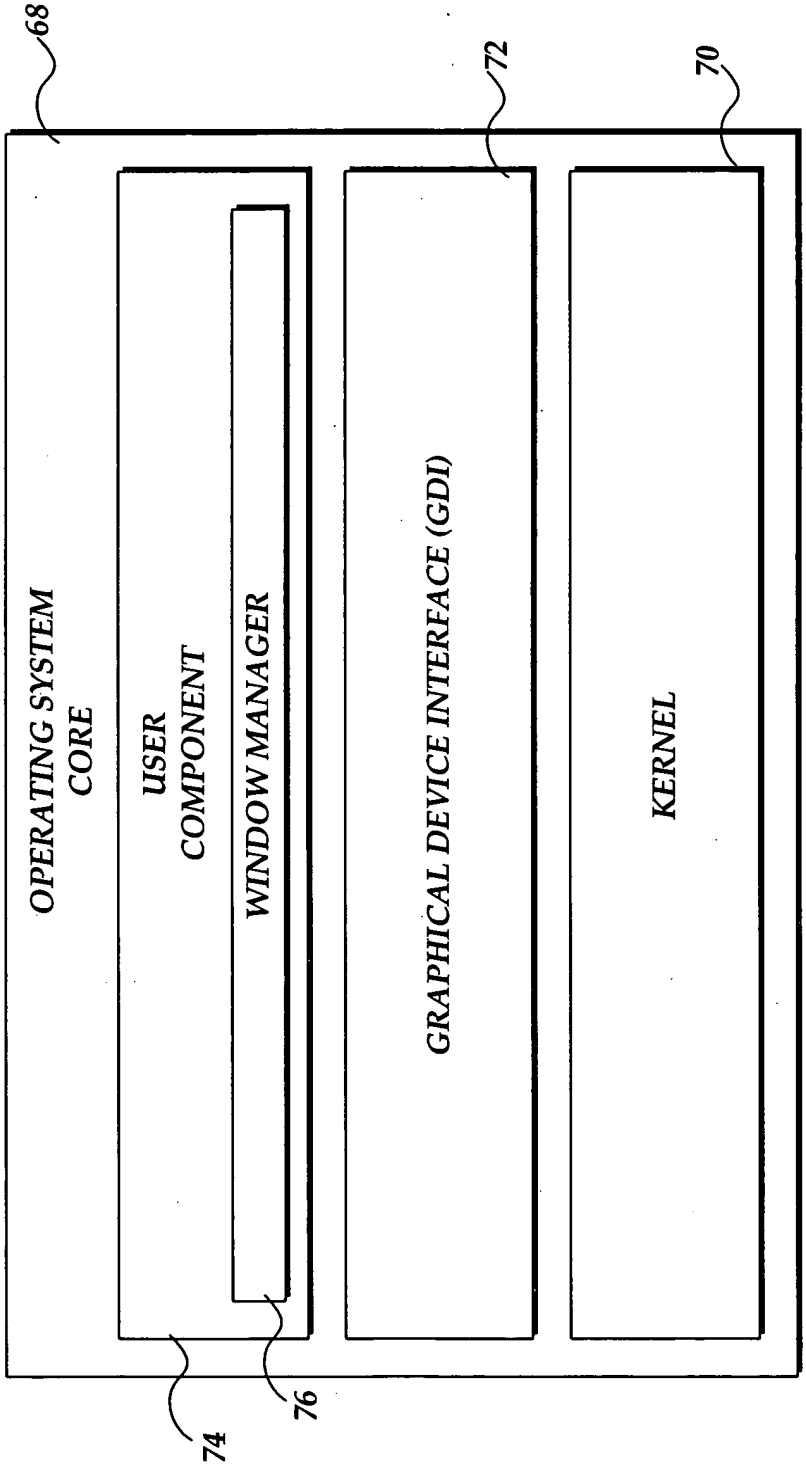
68
72
70
OPERATING SYSTEM CORE
USER COMPONENT
WINDOW MANAGER
GRAPHICAL DEVICE INTERFACE (GDI)
KERNEL
74
76

Fig.3

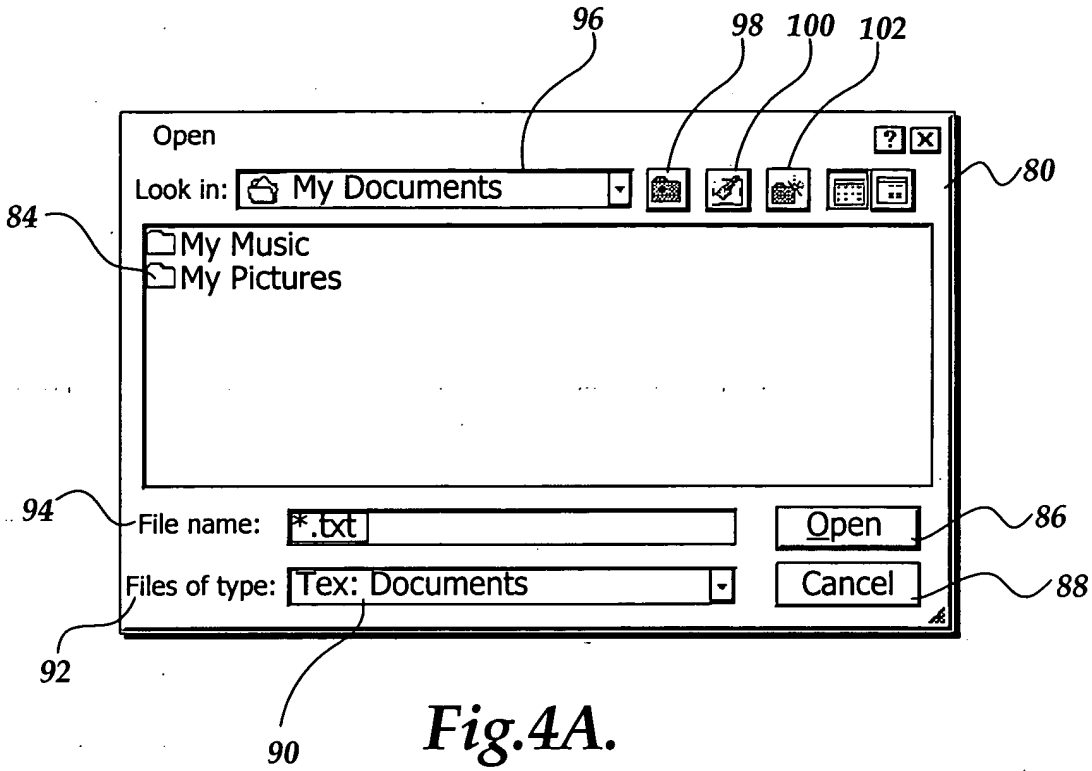
96
98
100
102
Open
Look in: My Documents
80
84
My Music
My Pictures
94
File name: *.txt
Open
86
Files of type: Tex: Documents
Cancel
88
92
90

Fig.4A.

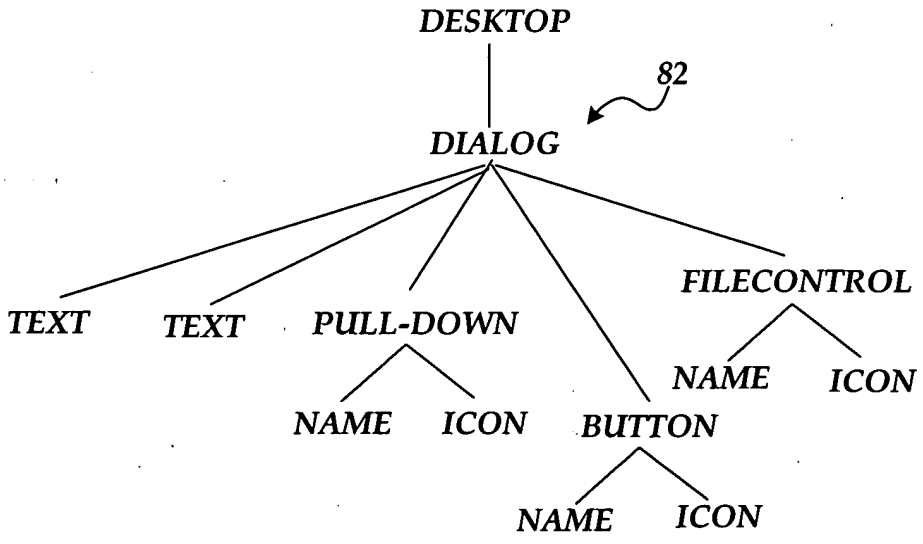
DESKTOP
82
DIALOG
TEXT
TEXT
PULL-DOWN
FILECONTROL
NAME
ICON
NAME
ICON
BUTTON
NAME
ICON

Fig.4B.

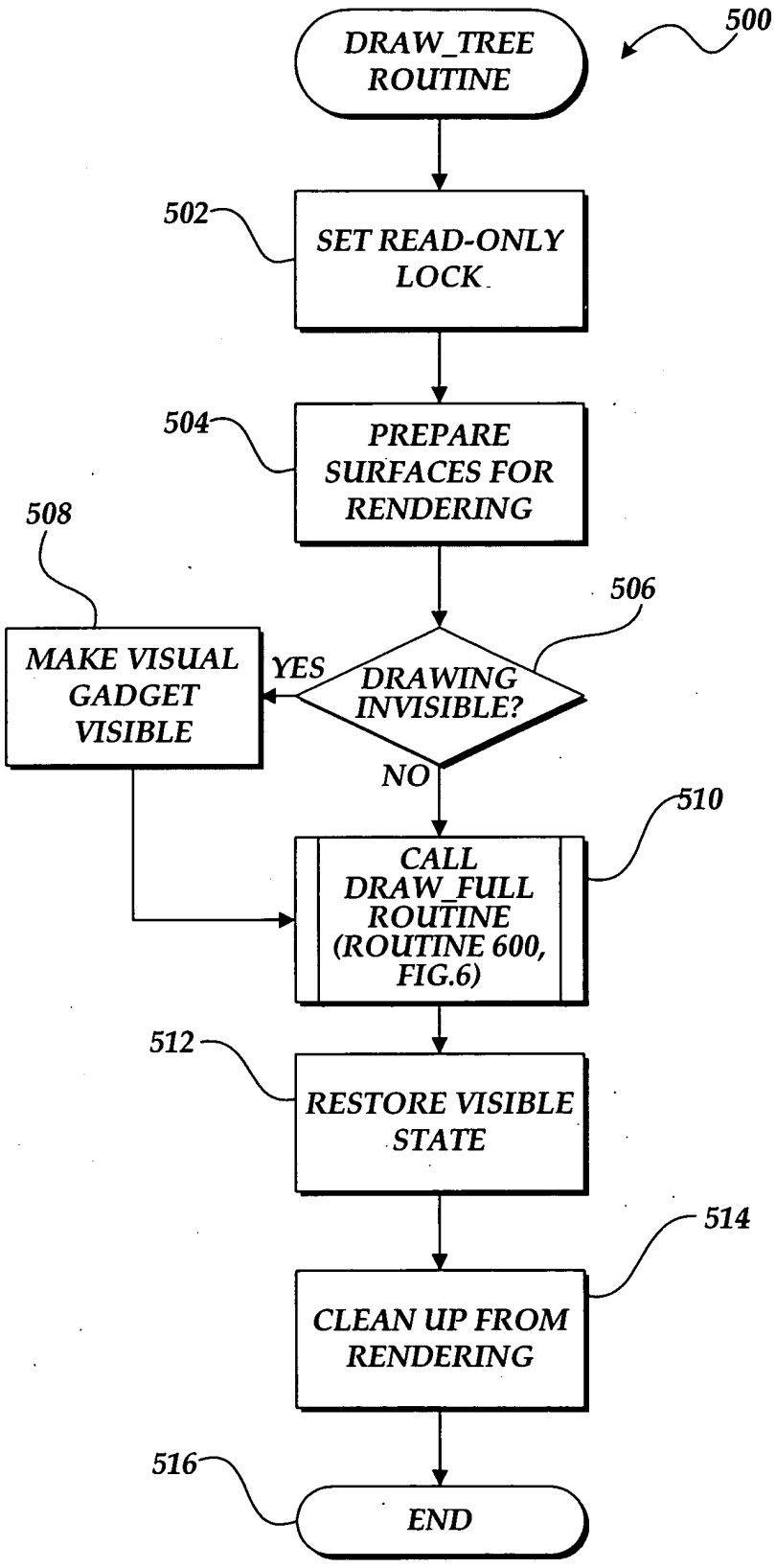
DRAW_TREE ROUTINE
500
SET READ-ONLY LOCK
502
PREPARE SURFACES FOR RENDERING
504
DRAWING INVISIBLE?
506
YES
MAKE VISUAL GADGET VISIBLE
508
NO
CALL DRAW_FULL ROUTINE (ROUTINE 600, FIG.6)
510
RESTORE VISIBLE STATE
512
CLEAN UP FROM RENDERING
514
END
516

Fig.5.

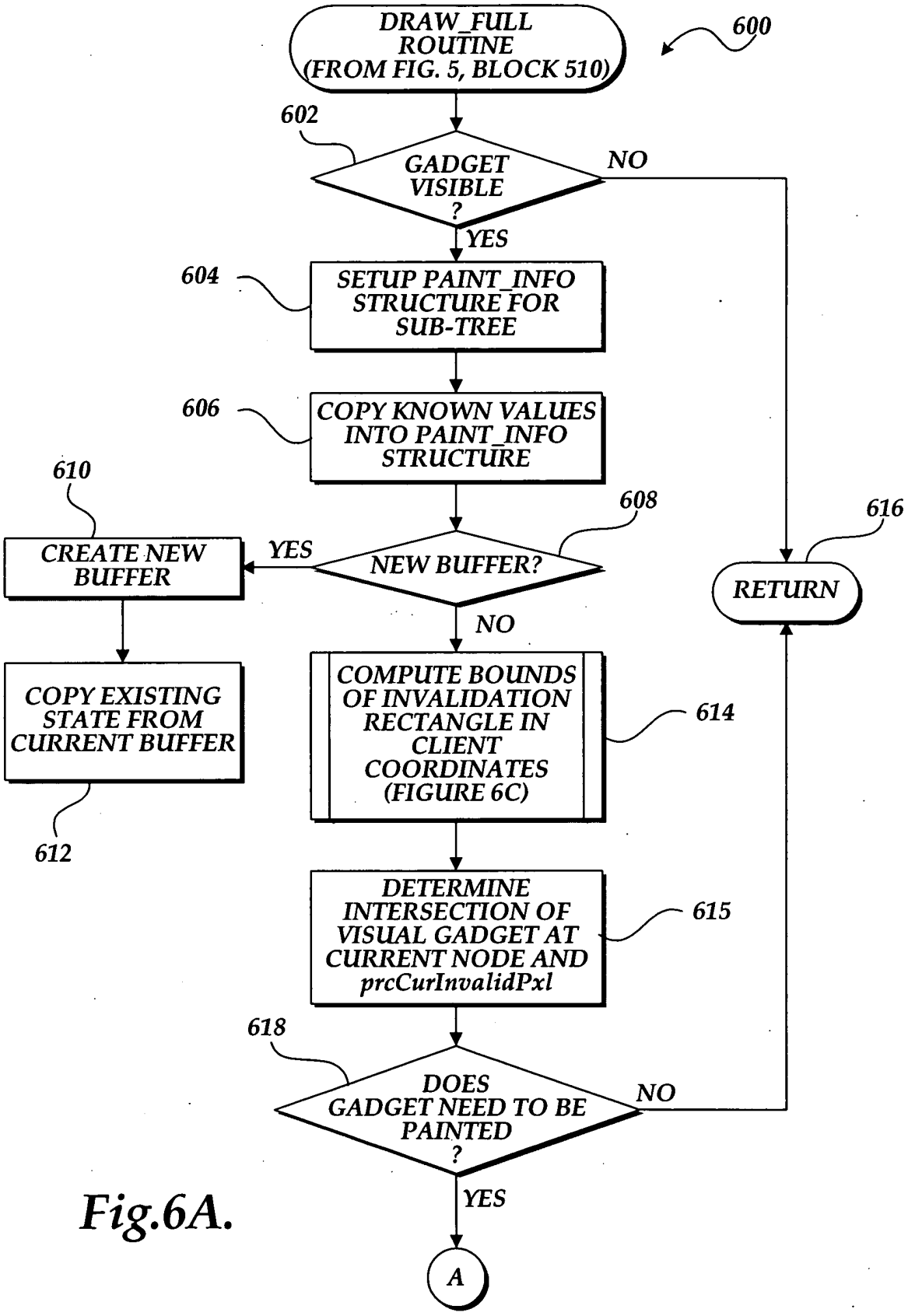
DRAW_FULL ROUTINE (FROM FIG. 5, BLOCK 510)
600
602
GADGET VISIBLE ?
NO
YES
604
SETUP PAINT_INFO STRUCTURE FOR SUB-TREE
606
COPY KNOWN VALUES INTO PAINT_INFO STRUCTURE
608
NEW BUFFER?
YES
610
CREATE NEW BUFFER
612
COPY EXISTING STATE FROM CURRENT BUFFER
NO
614
COMPUTE BOUNDS OF INVALIDATION RECTANGLE IN CLIENT COORDINATES (FIGURE 6C)
615
DETERMINE INTERSECTION OF VISUAL GADGET AT CURRENT NODE AND prcCurInvalidPxl
618
DOES GADGET NEED TO BE PAINTED ?
NO
YES
A
616
RETURN

Fig.6A.

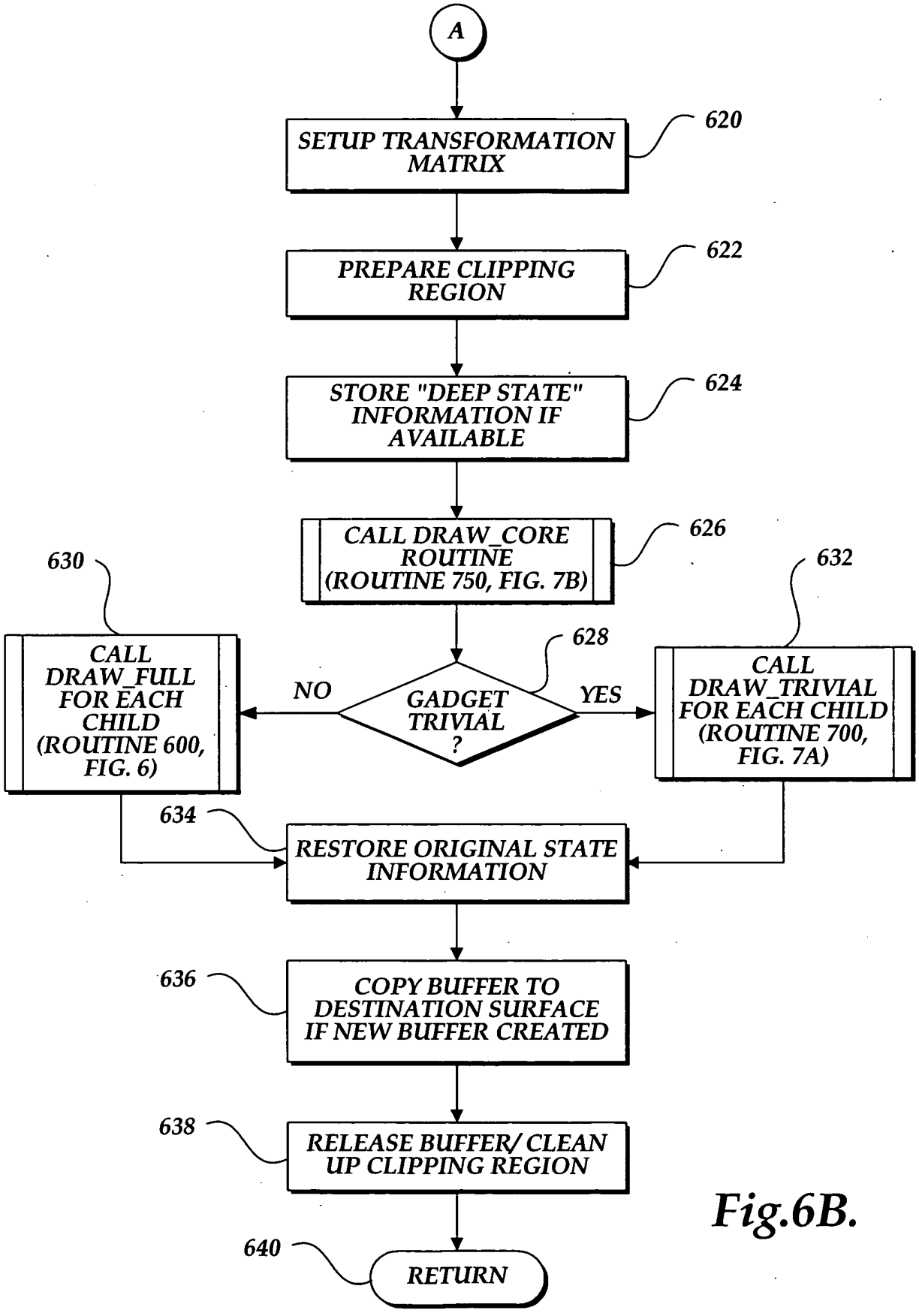
A
SETUP TRANSFORMATION MATRIX
620
PREPARE CLIPPING REGION
622
STORE "DEEP STATE" INFORMATION IF AVAILABLE
624
CALL DRAW_CORE ROUTINE (ROUTINE 750, FIG. 7B)
626
630
632
628
CALL DRAW_FULL FOR EACH CHILD (ROUTINE 600, FIG. 6)
NO
GADGET TRIVIAL ?
YES
CALL DRAW_TRIVIAL FOR EACH CHILD (ROUTINE 700, FIG. 7A)
634
RESTORE ORIGINAL STATE INFORMATION
636
COPY BUFFER TO DESTINATION SURFACE IF NEW BUFFER CREATED
638
RELEASE BUFFER/ CLEAN UP CLIPPING REGION
640
RETURN

Fig.6B.

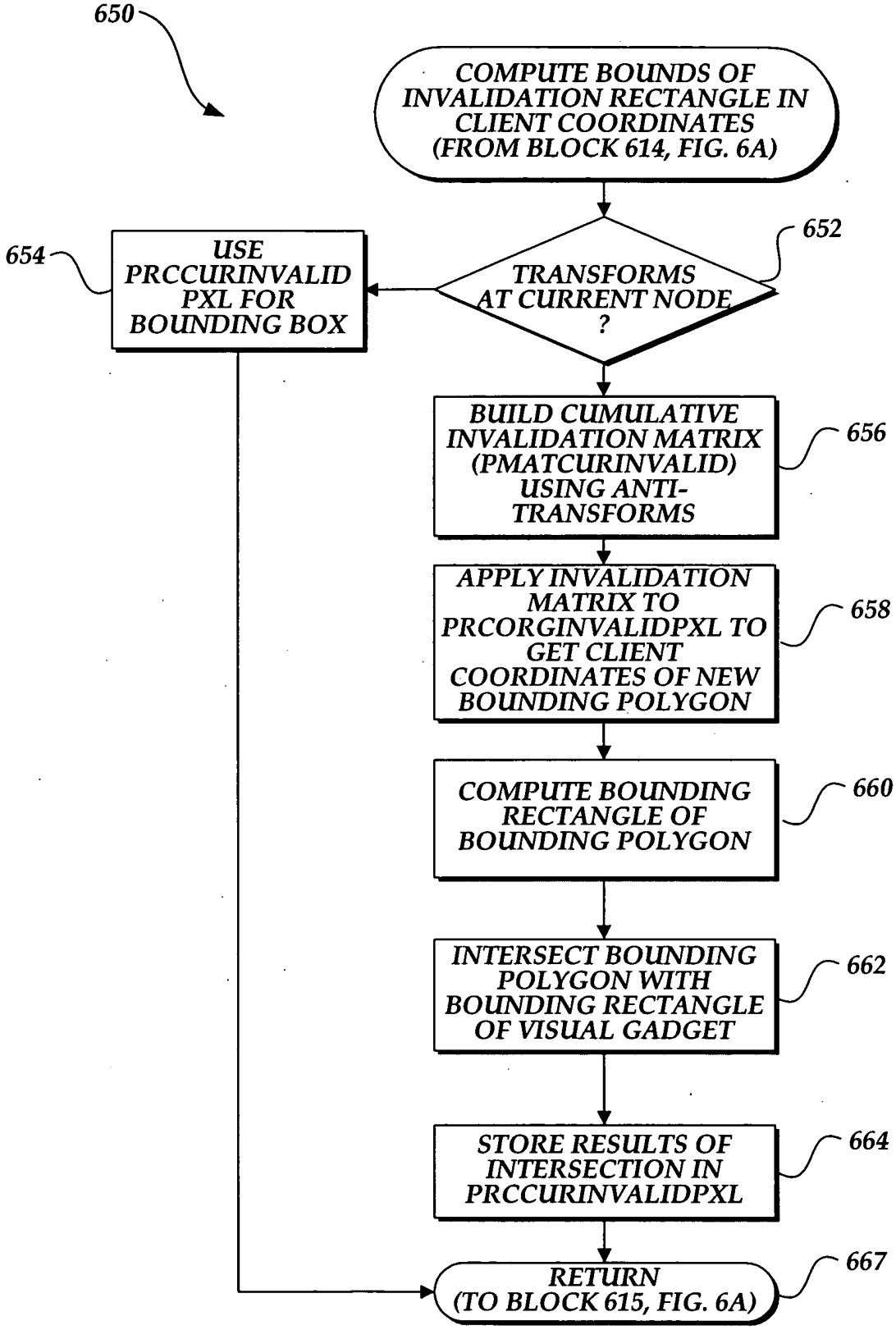
650
COMPUTE BOUNDS OF INVALIDATION RECTANGLE IN CLIENT COORDINATES (FROM BLOCK 614, FIG. 6A)
652
TRANSFORMS AT CURRENT NODE ?
654
USE PRCCURINVALID PXL FOR BOUNDING BOX
656
BUILD CUMULATIVE INVALIDATION MATRIX (PMATCURINVALID) USING ANTI-TRANSFORMS
658
APPLY INVALIDATION MATRIX TO PRCORGINVALIDPXL TO GET CLIENT COORDINATES OF NEW BOUNDING POLYGON
660
COMPUTE BOUNDING RECTANGLE OF BOUNDING POLYGON
662
INTERSECT BOUNDING POLYGON WITH BOUNDING RECTANGLE OF VISUAL GADGET
664
STORE RESULTS OF INTERSECTION IN PRCCURINVALIDPXL
667
RETURN (TO BLOCK 615, FIG. 6A)

Fig.6C.

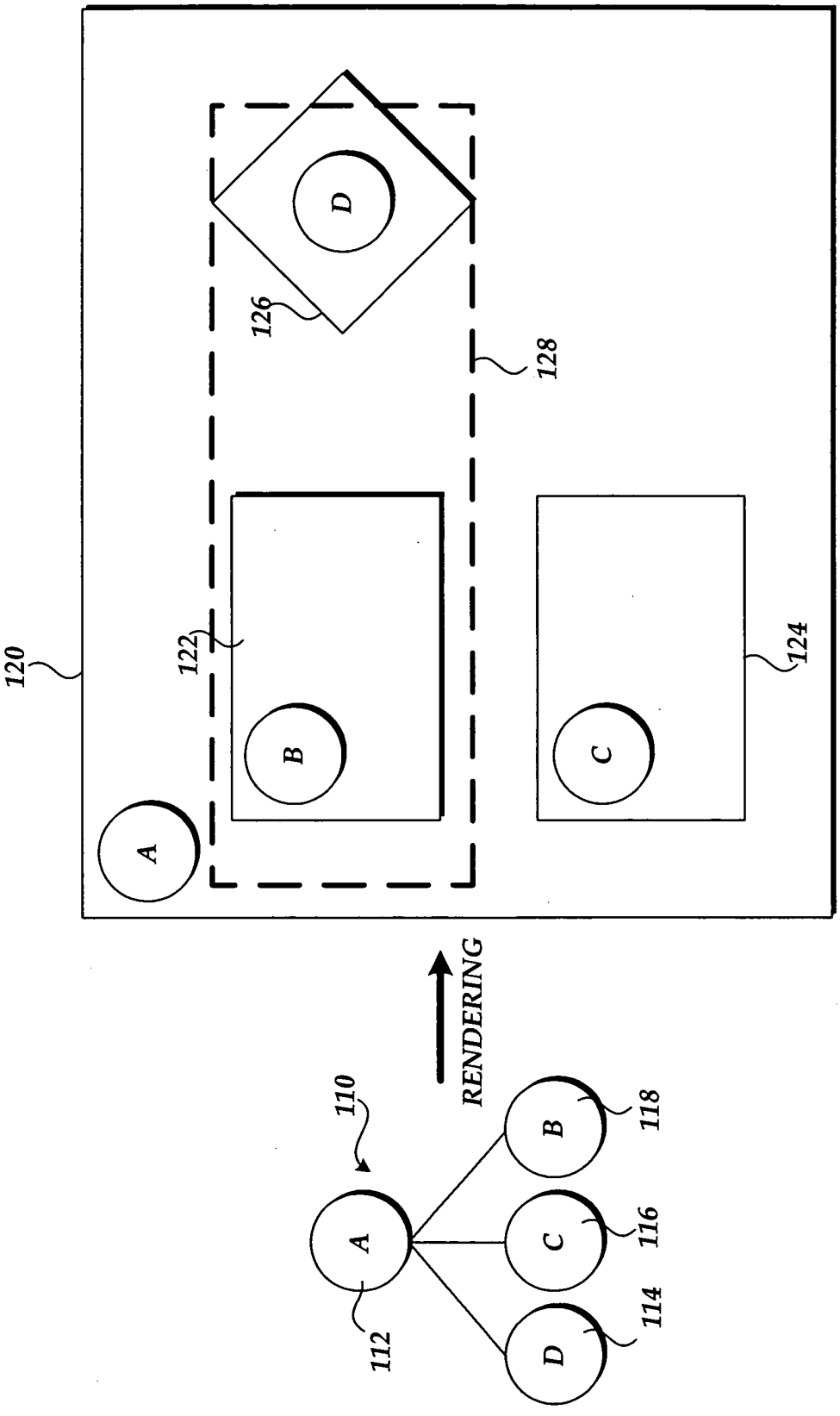

Fig.6D.

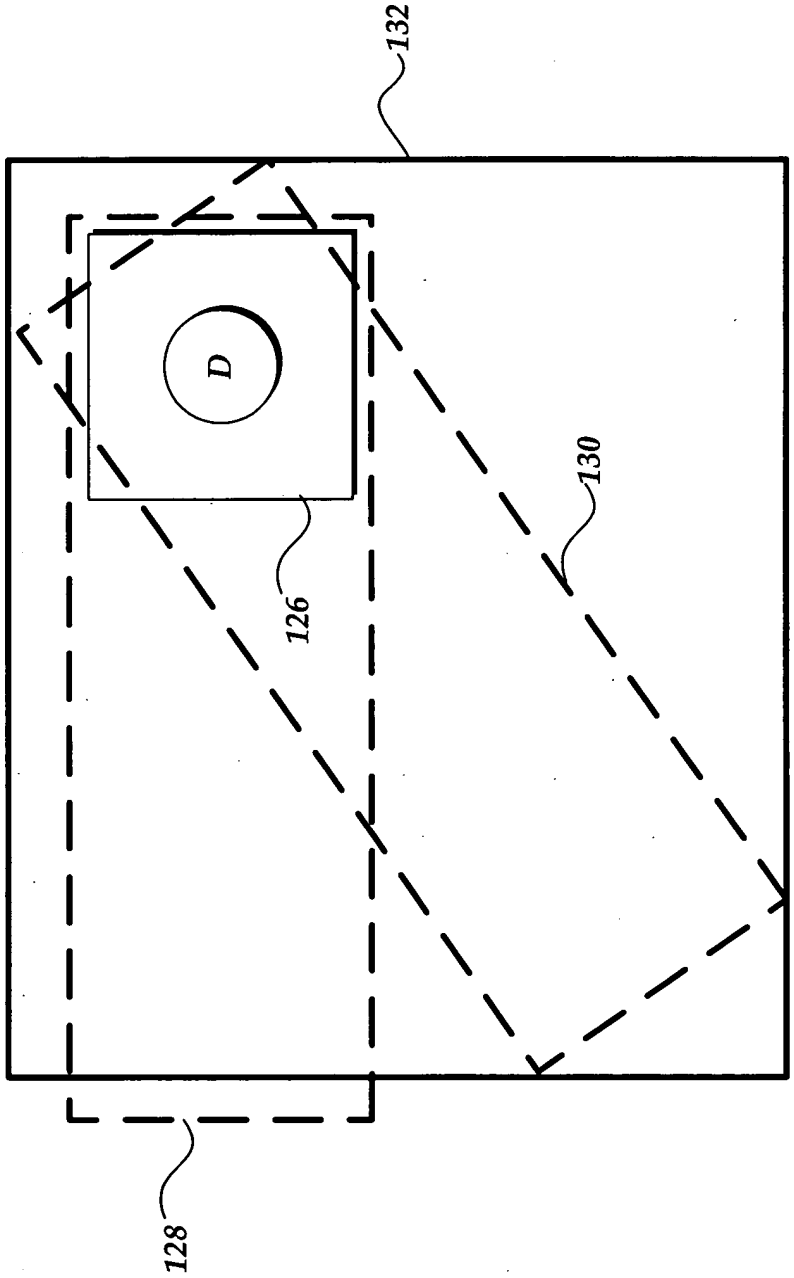

Fig.6E.

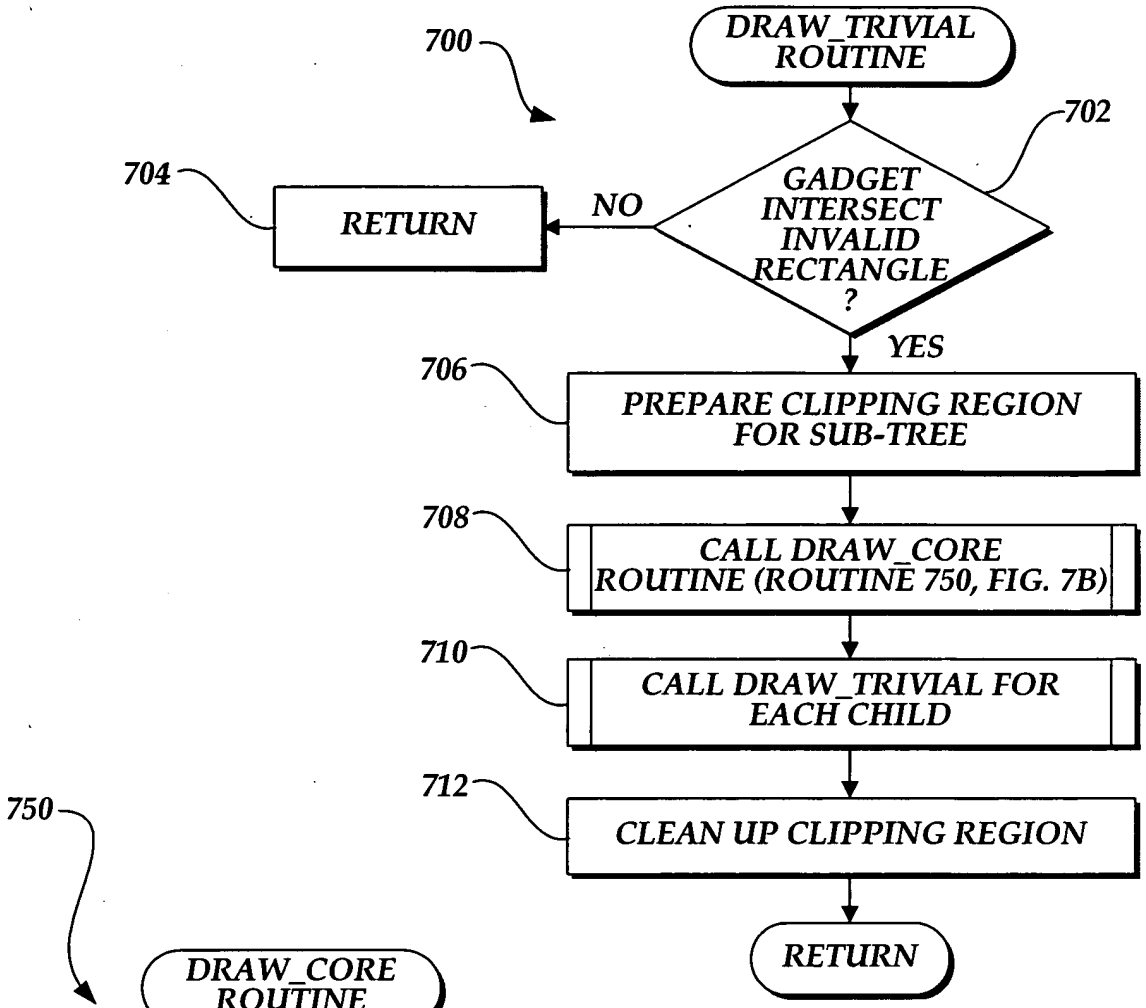
700
DRAW_TRIVIAL ROUTINE
702
GADGET INTERSECT INVALID RECTANGLE ?
NO
704
RETURN
YES
706
PREPARE CLIPPING REGION FOR SUB-TREE
708
CALL DRAW_CORE ROUTINE (ROUTINE 750, FIG. 7B)
710
CALL DRAW_TRIVIAL FOR EACH CHILD
712
CLEAN UP CLIPPING REGION
RETURN

Fig.7A.

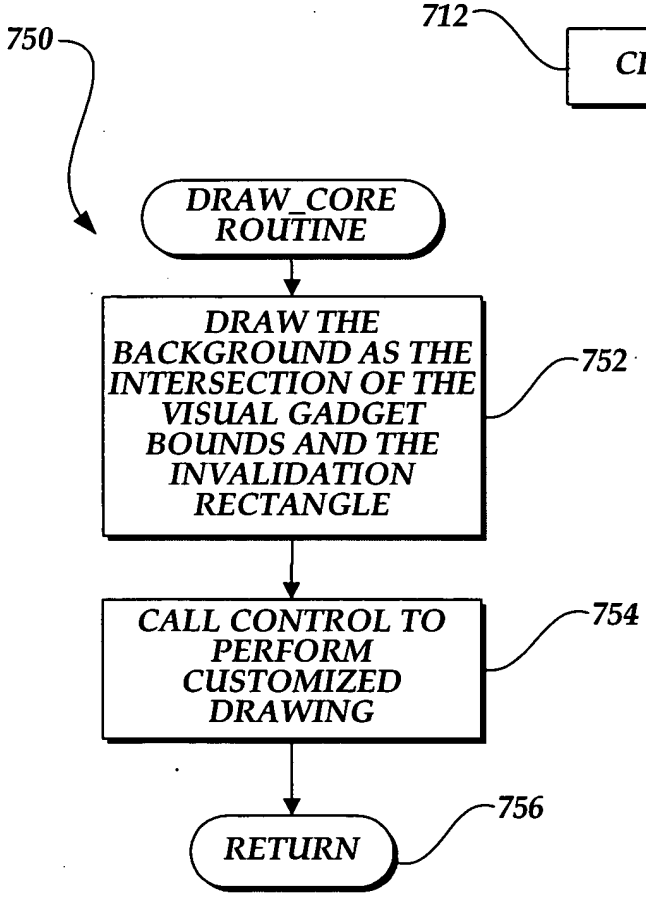
750
DRAW_CORE ROUTINE
DRAW THE BACKGROUND AS THE INTERSECTION OF THE VISUAL GADGET BOUNDS AND THE INVALIDATION RECTANGLE
752
CALL CONTROL TO PERFORM CUSTOMIZED DRAWING
754
RETURN
756

Fig.7B.

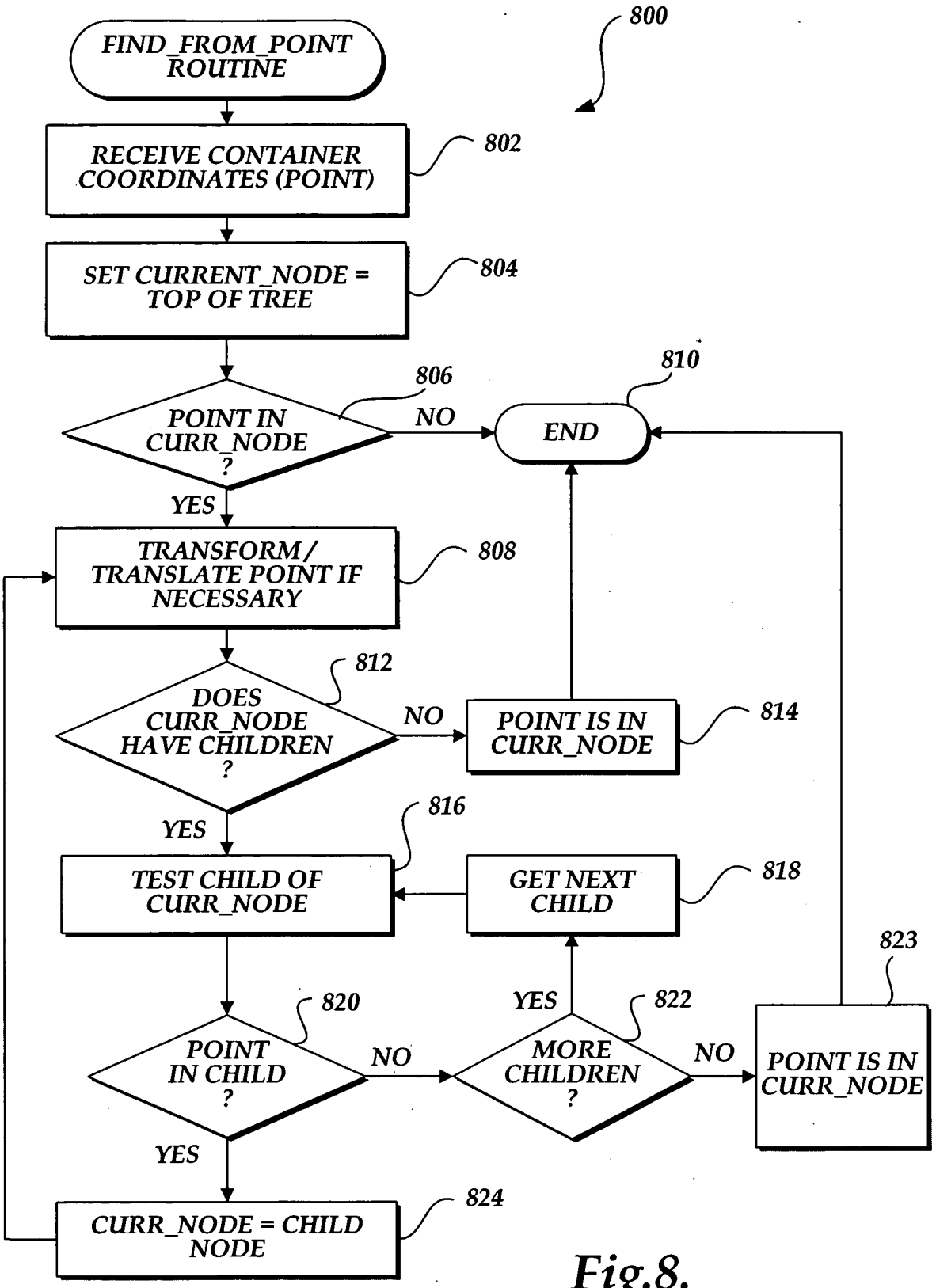
FIND_FROM_POINT ROUTINE
800
RECEIVE CONTAINER COORDINATES (POINT)
802
SET CURRENT_NODE = TOP OF TREE
804
POINT IN CURR_NODE ?
806
NO
END
810
YES
TRANSFORM/ TRANSLATE POINT IF NECESSARY
808
DOES CURR_NODE HAVE CHILDREN ?
812
NO
POINT IS IN CURR_NODE
814
YES
TEST CHILD OF CURR_NODE
816
GET NEXT CHILD
818
POINT IN CHILD ?
820
NO
YES
MORE CHILDREN ?
822
NO
POINT IS IN CURR_NODE
823
YES
CURR_NODE = CHILD NODE
824

Fig.8.

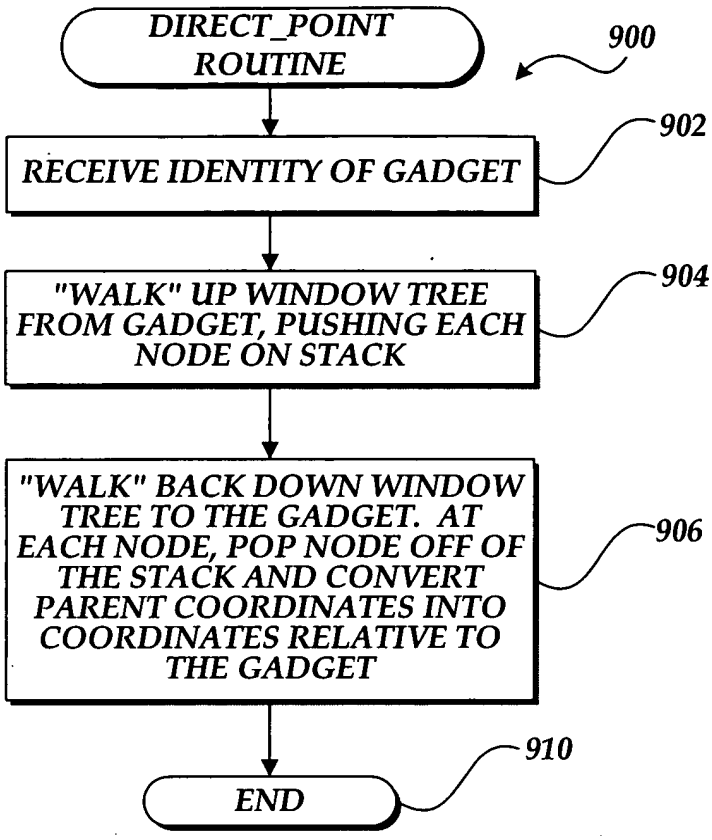
DIRECT_POINT ROUTINE
900
RECEIVE IDENTITY OF GADGET
902
"WALK" UP WINDOW TREE FROM GADGET, PUSHING EACH NODE ON STACK
904
"WALK" BACK DOWN WINDOW TREE TO THE GADGET. AT EACH NODE, POP NODE OFF OF THE STACK AND CONVERT PARENT COORDINATES INTO COORDINATES RELATIVE TO THE GADGET
906
END
910

Fig.9.

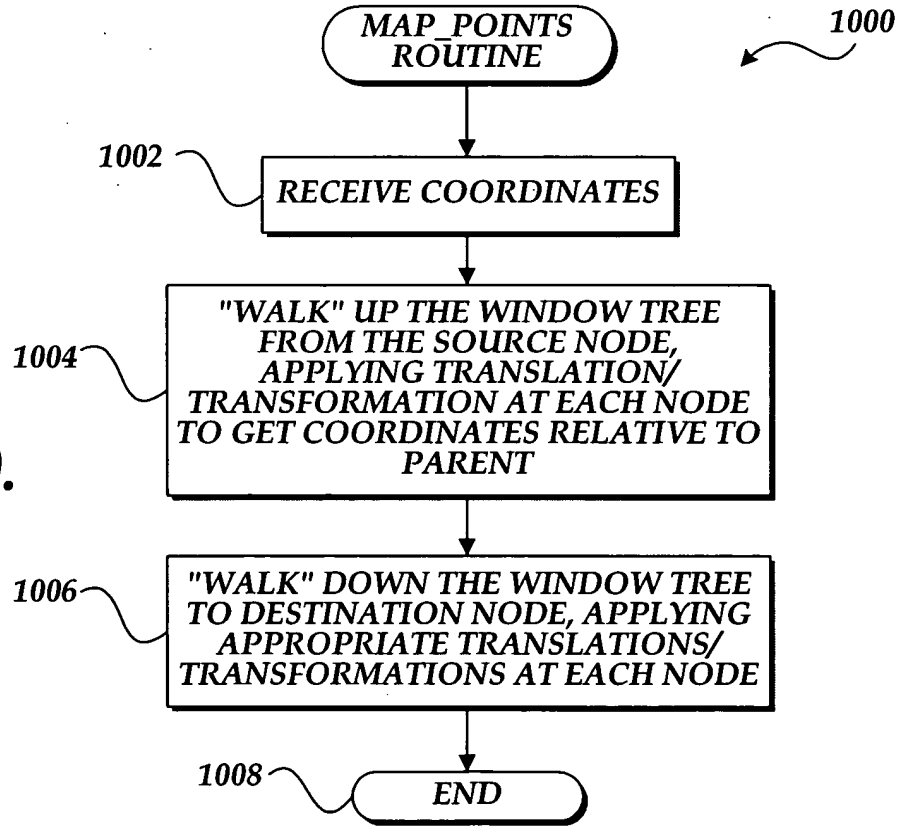
MAP_POINTS ROUTINE
1000
1002
RECEIVE COORDINATES
1004
"WALK" UP THE WINDOW TREE FROM THE SOURCE NODE, APPLYING TRANSLATION/ TRANSFORMATION AT EACH NODE TO GET COORDINATES RELATIVE TO PARENT
1006
"WALK" DOWN THE WINDOW TREE TO DESTINATION NODE, APPLYING APPROPRIATE TRANSLATIONS/ TRANSFORMATIONS AT EACH NODE
1008
END

Fig.10.